ERDC/ITL TR-01-5

Information Technology Laboratory

US Army Corps
of Engineers®
Engineer Research and
Development Center

Development of Geodetic Conversion Routines

Vaiyapuri Danushkodi

September 2001

ERDC/ITL TR-01-5
September 2001

Development of Geodetic Conversion Routines

by Vaiyapuri Danushkodi

Information Technology Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final report

Contents

Preface

The Geodetic Conversion Routines Report is a product of the Computer-Aided Design and Drafting/Geographic Information Systems (CADD/GIS) Technology Center for Facilities, Infrastructure, and Environment (Center), Information Technology Laboratory (ITL), U.S. Army Engineer Research and Development Center (ERDC) Project 01.008. The project was funded and executed by the Center. The Center was chartered in 1992 to promote the use of CADD and GIS technologies for life-cycle facilities management within the U.S. Army Corps of Engineers (CE), Navy, and Air Force. The Center operates under the guidance of Mr. Timothy D. Ables, Acting Director, ITL, and Mr. Harold Smith, Chief, CADD/GIS Technology Center. The Center functions under the guidance of several oversight committees including the Board of Directors, Corporate Staff, and several Field User Groups (FUG). The Civil Works FUG served as the project sponsor and provided technical guidance for the project. Members of these groups are listed below.

Board of Directors

Name	Membership	Affiliation[1]
Gary Erickson	Chair	DoD/Air Force
William Patrick Layne	Member	Coast Guard
John Nerger	Member	DoD/Army
Dwight Beranek	Member	DoD/CE
Dr. Lewis “Ed” Link	Member	DoD/CE
Paul Hubbell	Member	DoD/Marines
Dr. Get Moy	Member	DoD/Navy
Dr. John G. Lyon	Member	EPA
Kay McNew	Member	GSA
David Harris	Member	NIBS
Joseph Toussaint	Member	State
Harold Smith	Advisor	Center

[1] DoD = Department of Defense; EPA = Environmental Protection Agency; GSA = General Services Administration; NIBS = National Institute of Building Sciences.

Corporate Staff Membership

Name	Membership	Affiliation[1]
Jean McGinn	Chair	DoD/CE
Harold Smith	Member	Center
Paul Herold	Member	Coast Guard
Ronald S. (Stan) Gross	Member	DoD/Air Force
William A. Myers	Member	DoD/Air Force
Vicki Williams	Member	DoD/Air Force
Roderick Chisholm	Member	DoD/Army
Stan Shelton	Member	DoD/Army
Thomas Hart	Member	DoD/CE
Joseph Jaccobbazzi	Member	DoD/CE
M. K. Miles	Member	DoD/CE
Tony Vajda	Member	DoD/CE
Thomas M. Karst	Member	DoD/DCMA
Daniel McLaughlin	Member	DoD/DLA
Paul Bouley	Member	DoD/Marines
Bobby Bean	Member	DoD/Navy
Jim Carberry	Member	DoD/Navy
Chrisopher Kyburg	Member	DoD/Navy
Carolyn Wilber	Member	DoD/Navy
Wanda Hobart	Member	GSA
Earle Kennett	Member	NIBS
Jim Whittaker	Member	OSD
Robert E. Clarke	Member	State
Albert Johnson	Member	NASA
Stan Shelton	Member	DoD/Army
Linda Smith	Alternate	DoD/Army
Fredrik (Rik) Wiant	Alternate	DoD/CE
Steven Coppedge	Alternate	DoD/Navy
Francois Grobler	Advisor	DoD/CE
Richard A Hermann	Advisor	DoD/CE
Brian T. Tracy	Advisor	DoD/CE
Wade West	Advisor	DoD/CE
Gary Biggers	Advisor	DoD/Navy
Richard Herrmann	Advisor	DoD/CE
Andrew Brucewicz	Former Advisor	DoD/CE

[1] DCMA = Defense Contract Management Agency; DLA = Defense Logistics Agency; OSD = Office of the Secretary of Defense; NASA = National Aeronautics and Space Administration.

Civil Works Field Users Group		
Name	**Membership**	**Affiliation**
Doug Wall	Chair	DoD/CE
Randall Mayne	Vice-Chair	DoD/CE
Eugene Batty	Member	DoD/CE
Gerald Bowles	Member	DoD/CE
Terry Dawson	Member	DoD/CE
Blaise Grden	Member	DoD/CE
David Hawley	Member	DoD/CE
Tony Hill	Member	DoD/CE
Paul Murawsky	Member	DoD/CE
Ron Santos	Member	DoD/CE
Ralph Scheid	Member	DoD/CE
Mike Watson	Member	DoD/CE
Dr. V. Danushkodi	Facilitator	Center
Nancy Towne	Facilitator	Center

The Topographic Engineering Center, ERDC, provided a copy of the source code for CORPSCON, which helped the Center to build the SEMMSCON.DLL library. Dr. Barry McCleave of ITL provided programming consultations. Mike Grounds of Beacon Resources provided a SEMMSCON.LIB file, header information for function calls, and three example problems for use. Mr. Grounds also provided a non-CORPSCON routine for geodetic conversion and source code for a Visual Basic program that can be used in a Palm Pilot. The non-CORPSCON routine was beyond the scope of the current study and not included in this report. Mr. Grounds may be contacted at *mike@riversrus.com* for more information.

At the time of publication of this report, Director of ERDC was Dr. James R. Houston. Commander and Executive Director was COL John W. Morris III, EN.

1 Introduction

Several horizontal control datums exist for monument data, e.g., North American Datum (NAD) of 1927, NAD of 1983, State Plane Coordinate System (SPCS) of 1927, SPCS of 1983, Universal Tranverse Mercator (UTM) of 1927, and UTM of 1983. Vertical controls may be in North American Vertical Datum (NAVD) of 1988 or National Geodetic Vertical Datum (NGVD) of 1929. The National Geodetic Survey (NGS) developed algorithms for geodetic data conversions from one NAD to another NAD by a FORTRAN software known as NADCON. NGS also developed algorithms to convert elevation data between vertical datums in North America by another FORTRAN software, VERTCON. The U.S. Army Engineer Research and Development Center's Topographic Engineering Center (TEC) incorporated the NGS FORTRAN programs, NADCON, VERTCON, and GEOID, into a C software known as CORPSCON and expanded the conversion capabilities to SPCS and UTM. CORPSCON facilitates conversions of data in survey feet, international feet, meters, or decimal degrees. A Dynamic Link Library (DLL) was generated from CORPSCON's C code during the development of the Survey Engineering Monumentation Management System (SEMMS) software development. The primary objective of the current project is to make the DLL available to Government software developers with example codes using the DLL in Visual Basic, C, and C++ platforms.

2 Data Requirements for the DLL

SEMMSCON.DLL requires an environment file, C:\semms.env, pointing to the directory location of the geodetic conversion data files. The environment file should contain the directory path of the data files. The following data files are required to be in the directory pointed to by semms.env:

Latitude shift data files:

- Alaska.las
- Conus.las
- Hawaii.las
- Prvi.las

Longitude shift data files:

- Alaska.los
- Conus.los
- Hawaii.los
- Prvi.los

Vertical datum shift data files

- Vertconc.94
- Vertcone.94
- Vertconw.94

The DLL is generally installed to the Windows System32 directory so that it is available for all Windows programs.

3 Contents of SEMMSCON.DLL

The export table of the DLL contains 35 public functions available for geodetic conversions (Table 1). Available units of measure for conversions by the DLL are given in Table 2. Tables 3 and 4 provide zone numbers for use to facilitate datum conversions in UTM and SPCS, respectively.

Table 1
Function Names and Use

Ordinal #	Function Name	Description
0000	geo27_to_geo83	Convert lat, long from NAD27 to lat, long in NAD83
0001	geo27_to_sp27	Convert lat, long from NAD27 to State Plane 27
0002	geo27_to_sp83	Convert lat, long from NAD27 to State Plane 83
0003	geo27_to_utm27	Convert lat, long from NAD27 to UTM27 East & North
0004	geo27_to_utm83	Convert lat, long from NAD27 to UTM83 East & North
0005	geo83_to_geo27	Convert lat, long from NAD83 to lat, long in NAD83
0006	geo83_to_sp27	Convert lat, long from NAD83 to State Plane 27 East & North
0007	geo83_to_sp83	Convert lat, long from NAD83 to State Plane 83 East & North
0008	geo83_to_utm27	Convert lat, long from NAD83 to UTM27 East & North
0009	geo83_to_utm83	Convert lat, long from NAD83 to UTM83 East & North
000a	init_geoid9396	Initialize Geoid in memory
000b	init_nadcon	Read NADCON datum shift files to memory
000c	init_vertcon	Read VERTCON datum shift files to memory
000d	navd88_to_ngvd29	Convert NAVD88 elevation to NGVD29 elevation
000e	ngvd29_to_navd88	Convert NGVD29 elevation to NAVD88 elevation
000f	p27_to_geo27	Convert State Plane 27 East & North to lat & long in NAD27
0010	sp27_to_geo83	Convert State Plane 27 East & North to lat & long in NAD83
0011	sp27_to_sp83	Convert State Plane 27 East & North to East & North in State Plane 83
0012	sp27_to_utm27	Convert State Plane 27 East & North to UTM83 East & North
0013	sp27_to_utm83	Convert State Plane East & North to UTM 83 East & North
0014	sp83_to_geo27	Convert State Plane 83 East & North to lat & long in NAD27

(Continued)

Table 1 (Concluded)

Ordinal #	Function Name	Description
0015	sp83_to_geo83	Convert State Plane 83 East & North to lat & long in NAD83
0016	sp83_to_sp27	Convert State Plane 83 East & North to State Plane 27 East & North
0017	sp83_to_utm27	Convert State Plane 83 East & North to UTM27 East & North
0018	sp83_to_utm83	Convert State Plane 83 East & North to UTM83 East & North
0019	utm27_to_geo27	Convert UTM27 East & North to lat & long in NAD27
001a	utm27_to_geo83	Convert UTM27 East & North to lat & long in NAD83
001b	utm27_to_sp27	Convert UTM27 East & North to State Plane 27 East & North
001c	utm27_to_sp83	Convert UTM27 East & North to State Plane 83 East & North
001d	utm27_to_utm83	Convert UTM27 East & North to UTM83 East & North
001e	utm83_to_geo27	Convert UTM83 East & North to lat & long in NAD27
001f	utm83_to_geo83	Convert UTM83 East & North to lat & long in NAD83
0020	utm83_to_sp27	Convert UTM83 East & North to State Plane 27 East & North
0021	utm83_to_sp83	Convert UTM83 East & North to State Plane 83 East & North
0022	utm83_to_utm27	Convert UTM83 East & North to UTM27 East & North

Table 2
Units of Measure

Number used in Function	Units of Measure
1	Survey feet
2	International feet
3	Meters

Table 3
UTM Zones

Range of Longitude Degrees	Zone	Range of Longitude Degrees	Zone
66W to 72W	19	120W to 126W	10
72W to 78W	18	126W to 132W	9
78W to 84W	17	132W to 138W	8
84W to 90W	16	138W to 144W	7
90W to 96W	15	144W to 150W	6
96W to 102W	14	150W to 156W	5
102W to 108W	13	156W to 162W	4
108W to 114W	12	162W to 168W	3
114W to 120W	11		

Table 4
SPCS Zones

State Plane Zone	Zone Number	State Plane Zone	Zone Number
Alabama East	101	Idaho Central	1102
Alabama West	102	Idaho East	1101
Alaska Zone 1	5001	Idaho West	1103
Alaska Zone 2	5002	Illinois East	1201
Alaska Zone 3	5003	Illinois West	1202
Alaska Zone 4	5004	Indiana East	1301
Alaska Zone 5	5005	Indiana West	1302
Alaska Zone 6	5006	Iowa North	1401
Alaska Zone 7	5007	Iowa South	1402
Alaska Zone 8	5008	Kansas North	1501
Alaska Zone 9	5009	Kansas South	1502
Alaska Zone 10	5010	Kentucky North	1601
Arizona Central	202	Kentucky South	1602
Arizona East	201	Louisiana North	1701
Arizona West	203	Louisiana South	1702
Arkansas North	301	Lousiana Offshore	1703
Arkansas South	302	Maine East	1801
California I	401	Maine West	1802
California II	402	Maryland	1900
California III	403	Massachusetts Island	2002
California IV	404	Massachusetts Mainland	2001
California V	405	Michigan Central (L)	2112
California VI	406	Michigan East	2101
California VII	407	Michigan North	2111
Colorado Central	502	Michigan South	2113
Colorado North	501	Michigan West	2103
Colorado South	503	Michigan Central (TM)	2102
Connecticut	600	Minnesota Central	2202
Delaware	700	Minnesota North	2201
Florida East	901	Minnesota South	2203
Florida West	902	Mississippi East	2301
Florida North	903	Mississippi West	2302
Georgia East	1001	Missouri Central	2402
Georgia West	1002	Missouri East	2401
Hawaii 1	5101	Missouri West	2403
Hawaii 2	5102	Montana Central (NAD27)	2502
Hawaii 3	5103	Montana North (NAD27)	2501
Hawaii 4	5104	Montana South (NAD27)	2503
Hawaii 5	5105	Nebraska North (NAD27)	2601

(Continued)

Table 4 (Concluded)

State Plane Zone	Zone Number	State Plane Zone	Zone Number
Nebraska South (NAD27)	2602	South Dakota South	4002
Nevada East	2701	St. Croix	5201
Nevada Central	2702	Tennessee	4100
Nevada West	2703	Texas North	4201
New Hampshire	2800	Texas North Central	4202
New Jersey	2900	Texas Central	4203
New Mexico East	3001	Texas South Central	4204
New Mexico Central	3002	Texas South	4205
New Mexico West	3003	Utah North	4301
New York Central	3102	Utah Central	4302
New York East	3101	Utah South	4303
New York West	3103	Vermont	4400
North Carolina	3200	Virginia North	4501
North Dakota North	3301	Virginia South	4502
North Dakota South	3302	Washington North	4601
Ohio North	3401	Washington South	4602
Ohio South	3402	West Virginia North	4701
Oklahoma North	3501	West Virginia South	4702
Oklahoma South	3502	Wisconsin North	4801
Oregon North	3601	Wisconsin Central	4802
Oregon South	3602	Wisconsin South	4803
Pennsylvania North	3701	Wyoming East	4901
Pennsylvania South	3702	Wyoming East Central	4902
Puerto Rico & Virgin Islands	5201	Wyoming West Central	4903
Rhode Island	3800	Wyoming West	4904
South Carolina North (NAD27)	3901	Montana (NAD83)	2500
South Carolina South (NAD27)	3902	Nebraska (NAD83)	2600
South Dakota North	4001	South Carolina (NAD83)	3900

Requirements to Use the DLL Functions

In any programming environment, the user should know the function usage basics, e.g., the function name, parameter names, and types of variables. It is also important to know whether variable values are directly used or only referenced.

Specific requirements and details for a Visual Basic application are provided in Appendix A. Similar details for C and C++ platforms are given in Appendices B and C, respectively.

Example Geodetic Conversion Problems

Example problelms are listed below:

- **Problem 1:** Convert given latitude and longitude in NAD83 to NAD27 State Plane coordinates, Easting and Northing.
  Input data: latitude = 33.520639, longitude = 91.204101, SPZone = 2302, output units survey feet.
- **Problem 2:** Convert elevation of a point in NAVD88 to NGVD29.
  Input data: The elevation of the point in Problem 1 is 100.00 survey feet (NAVD88).
- **Problem 3:** Convert NAD27 State Plane coordinates to NAD83 UTM coordinates.
  Input data: East SP = 564472.06, NorthSP = 972600.12 survey feet units, SPZone = 2401.

The source code in Visual Basic, input data, and output results are provided in Appendix A. Similar items are included for C and C++ platforms in Appendices B and C, respectively.

Comparison of Results

The example problems were solved first by using CORPSCON, which produced the output shown in Figures 1-3.

Figure 1. Example Problem 1 in CORPSCON

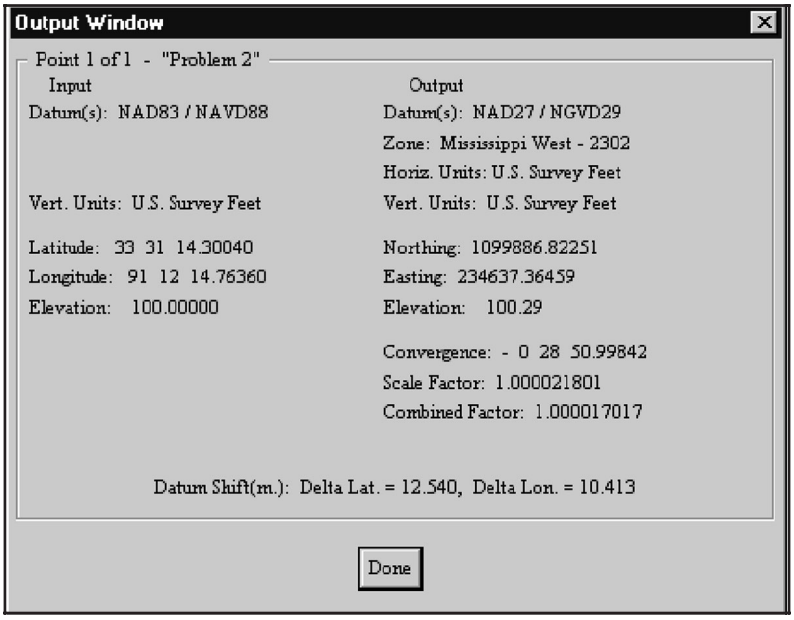

Figure 2. Example Problem 2 in CORPSCON

Figure 3. Example Problem 3 in CORPSCON

4 Summary of Results

Table 5 compares the results obtained from the Visual Basic, C, and C++ applications with those obtained from CORPSCON. The comparison shows that the DLL applications are working as expected. Additional information verifying the accuracy of CORPSCON may be obtained from *http://www.tec.army.mil*.

Table 5
Comparison of Example Problem Results with CORPSCON

Problem	Variable	CORPSCON	Visual Basic	C	C++
1	Northing	1099886.82	1099886.82	1099886.82	1099886.82
	Easting	234637.36	234637.36	234637.36	234637.36
2	Elevation (NGVD29)	100.29	100.29	100.29	100.29
3	UTM83 North	13993809.96	13993809.96	13993809.96	13993809.96
	UTM83 East	2420058.33	2420058.33	2420058.33	2420058.33

Appendix A
Using SEMMSCON.DLL in Visual Basic

The following code should be placed on top of a global module, prior to defining global variables or structures.

```
Public Declare Function init_nadcon Lib "semmscon.dll" () As Integer
Public Declare Function init_vertcon Lib "semmscon.dll" () As Integer
Public Declare Function init_geoid9396 Lib "semmscon.dll" () As Integer
Public Declare Sub geo83_to_utm83 Lib "semmscon.dll" (ByVal lat83 As Double, ByVal lon83
As Double, ByRef utm83x As Double, ByRef utm83y As Double, ByVal utm83_zone As Integer,
ByVal utm83_units As Integer)
Public Declare Sub geo27_to_geo83 Lib "semmscon.dll" (ByVal lat27 As Double, ByVal lon27
As Double, ByRef lat83 As Double, ByRef lon83 As Double)
Public Declare Sub geo83_to_geo27 Lib "semmscon.dll" (ByVal lat83 As Double, ByVal lon83
As Double, ByRef lat27 As Double, ByRef lon27 As Double)
Public Declare Sub geo27_to_utm83 Lib "semmscon.dll" (ByVal lat27 As Double, ByVal lon27
As Double, ByRef utm83x As Double, ByRef utm83y As Double, ByVal utm83_zone As Integer,
ByVal utm83_units As Integer)
Public Declare Sub geo83_to_utm27 Lib "semmscon.dll" (ByVal lat83 As Double, ByVal lon83
As Double, ByRef utm27x As Double, ByRef utm27y As Double, ByVal utm27_zone As Integer,
ByVal utm27_units As Integer)
Public Declare Sub geo83_to_sp83 Lib "semmscon.dll" (ByVal lat83 As Double, ByVal lon83 As
Double, ByRef utm27x As Double, ByRef utm27y As Double, ByVal utm27_zone As Integer, ByVal
utm27_units As Integer)
Public Declare Sub geo83_to_sp27 Lib "semmscon.dll" (ByVal lat83 As Double, ByVal lon83 As
Double, ByRef utm27x As Double, ByRef utm27y As Double, ByVal utm27_zone As Integer, ByVal
utm27_units As Integer)
Public Declare Sub navd88_to_ngvd29 Lib "semmscon.dll" (ByVal lat83 As Double, ByVal lon83
As Double, ByVal navd88z As Double, ByVal navd88_units As Integer, ByRef ngvd29z As
Double, ByVal ngvd29_units As Integer)
Public Declare Sub sp83_to_geo83 Lib "semmscon.dll" (ByVal sp83x As Double, ByVal sp83y As
Double, ByVal sp83zone As Integer, ByVal sp83units As Integer, ByRef lat83 As Double,
ByRef lon83 As Double)
Public Declare Sub sp27_to_geo83 Lib "semmscon.dll" (ByVal sp27x As Double, ByVal sp27y As
Double, ByVal sp27zone As Integer, ByVal sp27units As Integer, ByRef lat83 As Double,
ByRef lon83 As Double)
Public Declare Sub utm83_to_geo83 Lib "semmscon.dll" (ByVal utm83x As Double, ByVal utm83y
As Double, ByVal utm83zone As Integer, ByVal utm83units As Integer, ByRef lat83 As Double,
ByRef lon83 As Double)
Public Declare Sub utm27_to_geo83 Lib "semmscon.dll" (ByVal utm27x As Double, ByVal utm27y
As Double, ByVal utm83zone As Integer, ByVal utm83units As Integer, ByRef lat83 As Double,
ByRef lon83 As Double)
Public Declare Sub sp27_to_geo27 Lib "semmscon.dll" (ByVal sp27x As Double, ByVal sp27y As
Double, ByVal sp27zone As Integer, ByRef lat83 As Double, ByRef lon83 As Double)
```

```
Public Declare Sub utm27_to_geo27 Lib "semmscon.dll" (ByVal utm27x As Double, ByVal utm27y
As Double, ByVal utm83zone As Integer, ByVal utm83units As Integer, ByRef lat83 As Double,
ByRef lon83 As Double)
Public Declare Sub ngvd29_to_navd88 Lib "semmscon.dll" (ByVal lat83 As Double, lon83 As
Double, ByVal navd88z As Double, ByVal navdunits As Integer, ByRef ngvd29z As Double,
ByVal ngvd29_units As Integer)
Public Declare Sub sp27_to_utm83 Lib "semmscon.dll" (ByVal sp27x As Double, ByVal sp27y As
Double, ByVal sp27_zone As Integer, ByVal sp27_units As Integer, ByRef utm83x As Double,
ByRef utm83y As Double, ByVal utm83_zone As Integer, ByVal utm83_units As Integer)
```

The following source code should be copied to a form file attaching to a default command object:

Source Code to solve the three example problems:

```
Private Sub Command1_Click()
' Example 1: Convert from NAD83 lat/lon to SP27,Mississippi West, Survey ft.
Open "out.txt" For Output Access Write As #3
Dim Latitude As Double
Dim Longitude As Double
Dim Easting As Double
Dim Northing As Double
Dim ZoneNumber As Integer
Dim Units As Integer
Dim MsgSt1 As String
On Error Resume Next
Latitude = 33.520639
Longitude = 91.204101
ZoneNumber = 2302
Units = 1
' Initialize semmscon library
Dim i As Integer
i = init_nadcon()
i = init_vertcon()
i = init_geoid9396()
' Invoke conversion function for Example 1
Call geo83_to_sp27(Latitude, Longitude, Easting, Northing, ZoneNumber, Units)
Print #3, "Example problem 1: Convert lat/lon to SP27"
Print #3, "Input Lat=", Latitude, ",Lon=", Longitude, "Zone=", ZoneNumber, "Units=", Units
Print #3, "OutputNorthing=", Northing, "Easting=", Easting
MsgSt1 = "Example problem 1: Convert lat/lon to SP27" & vbCrLf
MsgSt1 = MsgSt1 & "Latitude=" & Latitude & vbCrLf
MsgSt1 = MsgSt1 & "Longitude=" & Longitude & vbCrLf
MsgSt1 = MsgSt1 & "ZoneNumber=" & ZoneNumber & vbCrLf
MsgSt1 = MsgSt1 & "Units=" & Units & vbCrLf
MsgSt1 = MsgSt1 & "Output Northing=" & Northing & vbCrLf
MsgSt1 = MsgSt1 & "Output Easting=" & Easting
MsgBox MsgSt1
' Example 2: Convert elevation from navd88 to ngvd29 datum
Dim Elev88 As Double
Dim Elev29 As Double
Dim UnitsIn As Integer
Dim UnitsOut As Integer
Elev88 = 100#
UnitsIn = 1
UnitsOut = 1
```

```
' Invoke elevation conversion function
Call navd88_to_ngvd29(Latitude, Longitude, Elev88, UnitsIn, Elev29, UnitsOut)
Print #3, "Problem 2: Convert Elevation from NAVD88 to NGVD29"
Print #3, "Input elev=", Elev88, "Elev29 = ", Elev29; ""
MsgSt1 = "Problem 2: Convert Elevation from NAVD88 to NGVD29" & vbCrLf
MsgSt1 = MsgSt1 & "Input Elev88=" & Elev88 & vbCrLf
MsgSt1 = MsgSt1 & "Output Elev29=" & Elev29
MsgBox MsgSt1
' Example 3:Convert from sp27 to utm83.
Dim EastSP As Double
Dim NorthSP As Double
Dim EastUTM As Double
Dim NorthUTM As Double
Dim Zone As Integer

EastSP = 564472.06
NorthSP = 972600.12
Zone = 2401
Units = 1
UTMZone = 15
Call sp27_to_utm83(EastSP, NorthSP, Zone, Units, EastUTM, NorthUTM, UTMZone, Units)
Print #3, "Problem 3: Convert from SP27 to UTM83"
Print #3, "Input EastSP=", EastSP, "NorthSP", NorthSP
Print #3, "OutputEastUTM=", EastUTM, "NorthUTM=", NorthUTM
MsgSt1 = "Problem 3: Convert from SP27 to UTM83" & vbCrLf
MsgSt1 = MsgSt1 & "Input EastSP=" & EastSP & vbCrLf
MsgSt1 = MsgSt1 & "Input NorthSP=" & NorthSP & vbCrLf
MsgSt1 = MsgSt1 & "Output EastUTM=" & EastUTM & vbCrLf
MsgSt1 = MsgSt1 & "Output NorthUTM=" & NorthUTM
MsgBox MsgSt1
Close #3
MsgSt1 = "Results of all examples are in file Out.txt"
MsgBox MsgSt1
End Sub
```

Execution of the Visual Basic application produces the solution screens shown in Figures A1-A5:

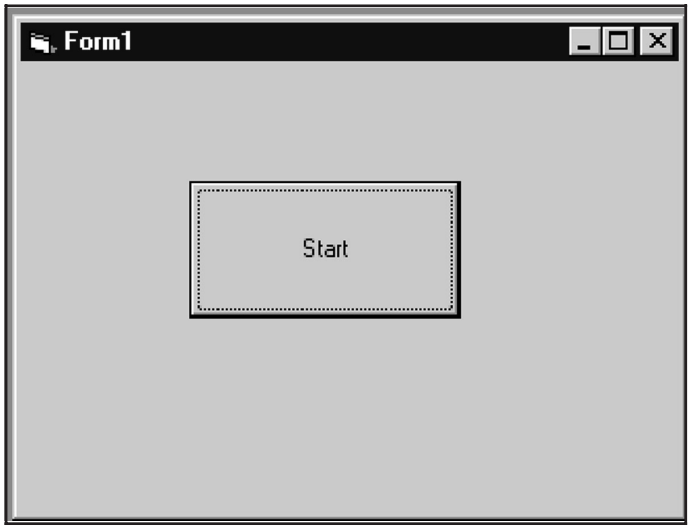

Figure A1. Output main screen. Click on the Start button to continue

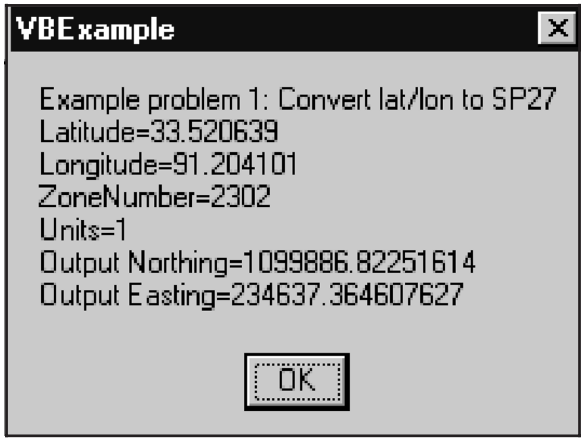

Figure A2. Solution screen for Problem 1. Click OK button to continue

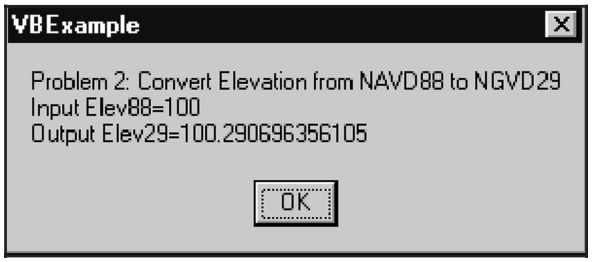

Figure A3. Solution screen for Problem 2. Click OK button to continue

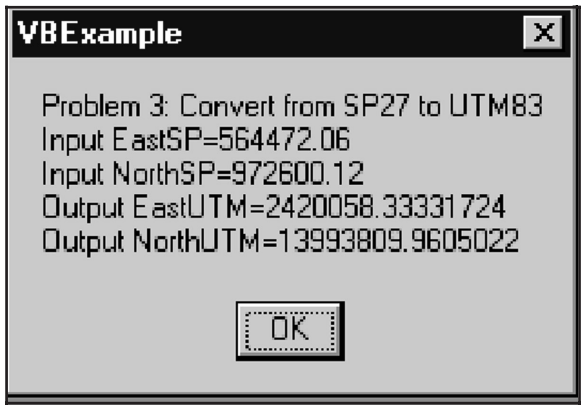

Figure A4. Solution screen for Problem 3. Click OK button to continue

```
Example problem 1: Convert lat/lon to SP27
Input Lat=     33.520639     ,Lon=          91.204101     Zone=          2302          Units=
OutputNorthing=              1099886.82251614            Easting=       234637.364607627
Problem 2: Convert Elevation from NAVD88 to NGVD29
Input elev=    100           Elev29 =       100.290696356105
Problem 3: Convert from SP27 to UTM83
Input EastSP=  564472.06     NorthSP        972600.12
OutputEastUTM=               2420058.33331724            NorthUTM=      13993809.9605022
```

Figure A5. Contents of `Out.txt` file showing solution for all three example problems are provided

Appendix B
Examples in C Platform

Source Code

The following source code solves the same three geodetic conversion problems in C. The source code is compiled and linked with an additional library SEMMSCON.LIB to provide linkages to the DLL functions. Still, the compiled program cannot function without the existence of the DLL.

Function prototypes are provided only for needed functions to solve the example problems. Usually, a header file specifying the prototypes for all public functions in the DLL and a corresponding LIB file should be provided by the owner of the DLL.

```
/Geodetic Conversion Examples in C
#include <iostream.h>
#include <stdio.h>
#include <windows.h>
#include <conio.h>
// Function protypes
int  init_geoid9396( );
int    init_nadcon( );
int    init_vertcon( );
void    geo83_to_sp27(double lat83, double lon83,
           double *sp27x, double *sp27y,
           short int sp27_zone, short int sp27_units);
void    navd88_to_ngvd29(double lat83, double lon83,
           double navd88z, short int navd88_units,
           double *ngvd29z, short int ngvd29_units);
void    sp27_to_utm83(double sp27x, double sp27y,
           short int sp27_zone, short int sp27_units,
           double *utm83x, double *utm83y,short int utm83_zone, short int utm83_units);
void Example_Problem_1();
void Example_Problem_2();
void Example_Problem_3();
```

```
void main() {
        // Run the three example problems
        Example_Problem_1();
        Example_Problem_2();
        Example_Problem_3();
        getch();
};

// Example problem definitions
void Example_Problem_1()
{
        int i,j,k;
        double lat, lon, Easting, Northing;
        short int ZoneNumber, Units;
        i = init_nadcon();
        j =  init_vertcon();
        k =  init_geoid9396();
        lon = 91.204101;
        lat = 33.520639;
        ZoneNumber = 2302;
        Units = 1;
        Easting =0;
        Northing =0;
        geo83_to_sp27(lat, lon,
                  &Easting,&Northing,
                   ZoneNumber, Units);
        printf( "** Example Problem 1: ** \n");
        printf( "Input Data: Latitude & Longitude = %#10.6f, %#10.6f\n", lat,lon);
        printf( "Zone & Units = %d, %d\n", ZoneNumber, Units);
        printf( "Output: Easting & Northing = %#9.2f, %#9.2f\n", Easting, Northing);


};
void Example_Problem_2()
{
        double lat, lon, Easting, Northing, navd88z, ngvd29z;
        short int ZoneNumber, Units;
        lon = 91.204101;
        lat = 33.520639;
        ZoneNumber = 2302;
        Units = 1;
        navd88z= 100.00;
        Easting =0;
        Northing =0;
        navd88_to_ngvd29(lat, lon, navd88z, Units, &ngvd29z, Units);
        printf( "**Example Problem 2** \n");
        printf( "Input Data:  Longitude & Latitude = %#5.2f, %#5.2f\n", lon, lat);
        printf( "SPCS Zone & Unit of Measure = %d, %d\n", ZoneNumber, Units);
        printf( "NAVD88 Elevation= %#7.3f", navd88z);
        printf( "Output NGVD29 Elevation= %#7.3f\n", ngvd29z);
};
void Example_Problem_3()
{
        double EastSP, NorthSP, sp27x,sp27y, utm83x, utm83y;
        short int sp27_zone, sp27_units, utm83_units, utm83_zone;
        EastSP= 564472.06;
        sp27x= EastSP;
        NorthSP= 972600.12;
        sp27y= NorthSP;
        sp27_zone= 2401;
        sp27_units= 1;
```

```
        utm83_units= 1;
        utm83_zone= 15;
        sp27_to_utm83( sp27x, sp27y, sp27_zone, sp27_units, &utm83x, &utm83y, utm83_zone,
utm83_units);

        printf( "**Example Problem 3** \n");
        printf( "Input Data:  EastSP & NorthSP = %#7.2f, %#7.2f\n", EastSP, NorthSP);
        printf( "SPCS Zone & Unit of Measure = %d, %d\n", sp27_zone, sp27_units);
        printf( "Output UTM83x & UTM83y= %#9.2f, %#9.2f\n", utm83x, utm83y);
}
```

Instructions for Compiling, Linking, and Executing the C Program

A Microsoft C++ 6.0 compiler was used at the CADD/GIS Technology Center for Facilities, Infrastructure, and Environment. A simple empty Windows Console project was first created. Then, the source code file, `Cexample.c`, and `SEMMSCON.LIB` files were added to the same project. The Build option of the compiler produced `Cexample.exe` file, which was later executed to produce the output shown in Figure B1.

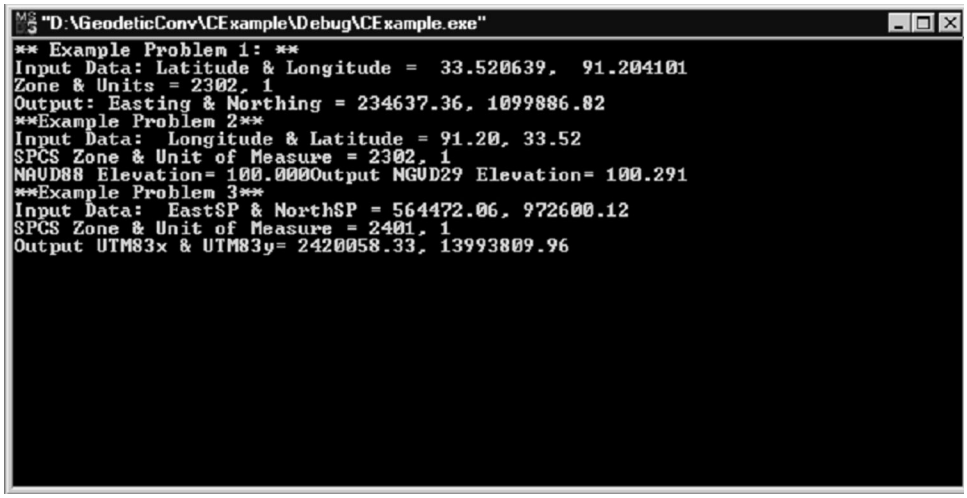

Figure B1. Cexample output screen

Appendix C
Example Problems in C++ Platform

The three example problems are solved using C++ code in this appendix. An empty Console Project was created in Microsoft Visual C++ 6.0 compiler. After adding the source code file, `CPPExample.cpp`, and `SEMMSCON.LIB` to the project, the Build option of the compiler was used to generate the `CPPExample.exe` file. The source code file contains the following:

```
//Geodetic Conversion Examples in C++


#include <iostream.h>
#include <stdio.h>
#include <windows.h>
#include <conio.h>
extern "C" _declspec( dllimport) int init_geoid9396( );
extern "C" _declspec( dllimport) int init_nadcon( );
extern "C" _declspec( dllimport) int init_vertcon( );
extern "C" _declspec( dllimport) void geo83_to_sp27(double , double ,
                  double *, double *,
                  short int , short int );
extern "C" _declspec( dllimport) void navd88_to_ngvd29(double, double,
                     double, short int,
                     double *, short int);
extern "C" _declspec( dllimport) void sp27_to_utm83(double, double,
short int, short,
                  double *, double *, short int, short int);

void Example_Problem_1 ();
void Example_Problem_2 ();
void Example_Problem_3 ();
void main() {
      // Read nadcon data files into memory
      int i = init_nadcon();
```

```
      // Read vertcon data files into memory
      int j =  init_vertcon();
      // Initialize Geoid in memory
      int k =  init_geoid9396();
      // Run three example problems
      Example_Problem_1 ();
      Example_Problem_2 ();
      Example_Problem_3 ();
      // Terminate program by a carriage return
      getch();
}


// Example Problem 1: Convert from NAD83 lat & lon to NAD27 State
Plane,Zone 2302 in Survey Feet
void Example_Problem_1 ()
{
      double lon = 91.204101;
      double lat = 33.520639;
      short int UtmZone =15;
      short int ZoneNumber = 2302;
      short int Units = 1;
      double  Easting =0;
      double  Northing =0;

      geo83_to_sp27(lat, lon,
                  &Easting,&Northing,
                   ZoneNumber, Units);
      char buf1[300] = " ";
          char s1[10] = " ";
            int j=0;
            j = sprintf( buf1, " Input Data: %s\n", s1);
            j += sprintf( buf1 +j, "\tLatitude: %s",s1 );
            j += sprintf( buf1 +j, "%#9.2f", lat);
            j += sprintf( buf1 +j, "  Longitude: %s",s1);
            j += sprintf( buf1 +j, "%#9.2f\n", lon);
            j += sprintf( buf1 +j, "  UTMzone: %s",s1);
            j += sprintf( buf1 +j, "%d", UtmZone);
            j += sprintf( buf1 +j, " InUnits: %s",  s1);
            j += sprintf( buf1 +j, "%d\n", Units);
            j += sprintf( buf1 +j, "Output Results: %s", s1);
            j += sprintf( buf1 +j, "East: %s", s1);
            j += sprintf( buf1 +j, "%#9.2f", Easting);
            j += sprintf( buf1 +j, "  North: %s", s1);
            j += sprintf( buf1 +j, "%#9.2f", Northing);
            printf("***Example Problem 1***: \n%s\n", buf1);
};
```

```
// Example Problem 2: Convert NAVD88 elev 100 ft to NGVD29 ft.
void Example_Problem_2 ()
{
      double lon = 91.204101;
      double lat = 33.520639;
      double navd88z =100.;
      double ngvd29z =0.0 ;
      short int Units = 1;
      navd88_to_ngvd29(lat, lon,
                       navd88z,  Units,
                      &ngvd29z, Units);
      char buf2[300] = " ";
          char s2[10] = " ";
            int j=0;
            j = sprintf( buf2, " Input Data: %s\n", s2);
            j += sprintf( buf2 +j, "\tLatitude: %s",s2 );
            j += sprintf( buf2 +j, "%#9.2f", lat);
            j += sprintf( buf2 +j, "  Longitude: %s",s2);
            j += sprintf( buf2 +j, "%#9.2f\n", lon);
            j += sprintf( buf2 +j, "  Elevation (NAVD88): %s",s2);
            j += sprintf( buf2 +j, "%#9.3f", navd88z);
            j += sprintf( buf2 +j, " InUnits: %s",  s2);
            j += sprintf( buf2 +j, "%d\n", Units);
            j += sprintf( buf2 +j, "Output Results: %s", s2);
            j += sprintf( buf2 +j, "Elevation(NGVD29): %s", s2);
            j += sprintf( buf2 +j, "%#9.3f", ngvd29z);
            j += sprintf( buf2 +j, "  Units: %s", s2);
            j += sprintf( buf2 +j, "%d", Units);
//Ge
            printf("***Example Problem 2***: \n%s\n", buf2);
};


// Example Problem 3:Convert NAD27 State Plane to NAD83 UTM Survey Feet.
void Example_Problem_3 ()
{
      double EastSP = 564472.06;
            double NorthSP = 972600.12;
            short int Zone = 2401;

            double lat83 = 0;
            double lon83 =0;
            short int Units = 1;

            short int utm83_zone = 15;
            double utm83x;
            double utm83y;
            double sp27x = 564472.06;
```

```
            double sp27y = 972600.12;
            short int sp27_zone =2401;
            short int sp27_units =1;
            short int utm83_units =1;
            sp27_to_utm83(sp27x, sp27y, sp27_zone,  sp27_units,
                  &utm83x, &utm83y, utm83_zone, utm83_units);

            char buf3[300] = " ";
          char s[10] = " ";
            int j=0;
            j = sprintf( buf3, " Input Data: %s\n", s);
            j += sprintf( buf3 +j, "\tInEastSP: %s",s );
            j += sprintf( buf3 +j, "%#9.2f", sp27x);
            j += sprintf( buf3 +j, "  InNorthSP: %s",s);
            j += sprintf( buf3 +j, "%#9.2f\n", sp27y);
            j += sprintf( buf3 +j, "  Insp_zone: %s",s);
            j += sprintf( buf3 +j, "%d", sp27_zone);
            j += sprintf( buf3 +j, " InUnits: %s",  s);
            j += sprintf( buf3 +j, "%d\n", sp27_units);
            j += sprintf( buf3 +j, "Output Results: %s", s);
            j += sprintf( buf3 +j, "UTM83x: %s", s);
            j += sprintf( buf3 +j, "%#9.2f", utm83x);
            j += sprintf( buf3 +j, "  UTM83y: %s", s);
            j += sprintf( buf3 +j, "%#9.2f", utm83y);
            printf("***Example Problem 3***: \n%s", buf3);
};
```

Execution of the `CPPExample.exe` file generates the solutions for the three example problems shown in Figure C1.

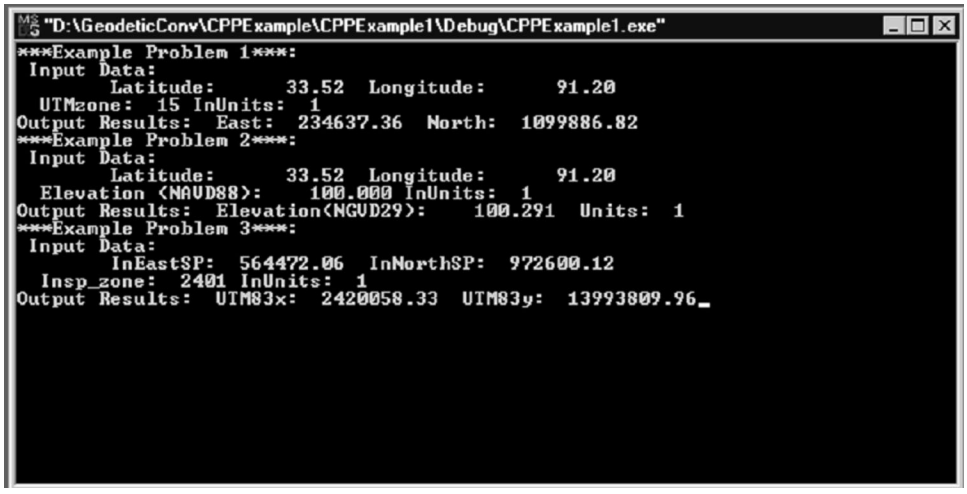

Figure C1. Output screen from C++ examples

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
September 2001	Final report	

4. TITLE AND SUBTITLE

Development of Geodetic Conversion Routines

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Vaiyapuri Danushkodi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Engineer Research and Development Center
Information Technology Laboratory
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER

ERDC/ITL TR-01-5

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Geodetic Conversion Routines Report is the final report of the Center Project 01.008. The report facilitates data conversion between various horizontal and vertical datums in the United States. It provides examples to invoke the Dynamic Link Library and use conversion functions in various programming languages, such as Visual Basic, C, and C++.

15. SUBJECT TERMS

Datum Data conversion NAD27 NAD83 NGVD29 NAVD DLL

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		28	

Destroy this report when no longer needed. Do not return it to the originator.